Daksh Pokar

dakshpokar@gmail.com | +1 217 (200)-2215 | dakshpokar.com | linkedin.com/in/dakshpokar | San Francisco, California, USA

Experience

Software Engineer **July 2025 – Present**
Meta Platforms, Inc. *Menlo Park, California, United States*

- Developed core features for Threads Ads from the ground up, taking the platform from 0 to 1.
- Drove **10% uplift in adoption** by integrating Threads into the Advantage+ bundle through Django & React updates.
- Boosted **adoption by 11%** by architecting the Static Carousel ad format across Hack, Python, Kotlin, Swift, and React.
- Reduced Dynamic Product Ads (DPA) **p99 latency by 60ms** on Threads by caching ads on user engagement.
- Led the full-stack development of Threads Reply Moderation by designing new Graph APIs and building the React frontend, which improved **revenue by 15%** and increased Threads Ads **adoption by 23%.**
- Increased **user engagement by 8%** by expanding ad aspect ratio support to 3:4, 4:5, and 9:16 formats.
- Reduced **resolution time by 30%** by LLM-powered Claude skill that automated alert triage and on-call incident workflows.
- Resolved critical SEVs and HiVAs through rapid root-cause analysis with internal debugging and logging tools.
- ***Skills:*** *Hack, Python, SQL, JavaScript, GraphQL, Kotlin, Swift, Thrift, Django, Claude Code, Jetpack Compose, A/B Testing*

Senior Software Engineer **Nov 2020 – Dec 2023**
Shipsy *Gurgaon, Haryana, India*

- Led the development of D2C Logistics-based SaaS products at Shipsy achieving over 1 million+ downloads.
- Improved system **reliability to 99.9%** uptime by architecting the decomposition of a monolithic service into three independent microservices — Payments, Booking, and Route Optimization.
- Reduced API **response times by 5%** by optimizing PostgreSQL queries, introducing Redis caching, and adding strategic indexes to MongoDB collections.
- Scaled the platform from **1k to 20k DAUs**, contributing to the company's growth and subsequent **$25M in funding.**
- Cut **deployment times by 75%** and eliminated 3 hours of manual intervention per release by implementing automated CI/CD pipelines using Jenkins.
- ***Skills:*** *Loopback.IO, NestJS, PostgreSQL, Docker, ElasticSearch, NodeJS, Flutter, ReactJS, MongoDB, TypeScript*

Technical Skills

Languages: Go, Hack, Python, JavaScript, TypeScript, Java, C++, Dart, SQL
Backend: NodeJS, ExpressJS, Django, LoopbackJS, NestJS, ElasticSearch, Flask, AWS (S3, ECS, Lambda)
Data: MySQL, MongoDB, PostgreSQL, Neo4J, Apache Spark, Apache Hadoop, Redis
Frontend & Mobile: ReactJS, NextJS, HTML, CSS | Flutter, Android, Jetpack Compose, ReactNative, Swift
Frameworks & Tools: Numpy, Pandas, Scikit-Learn, PyTorch, Langchain Keras, TensorFlow, Docker, Kubernetes, Helm Charts

Education

University of Illinois, Urbana-Champaign **Jan 2024 – May 2025**
Master of Computer Science — ***GPA: 4.0*** *Champaign, Illinois, United States*
Coursework: Cloud Computing Apps., Distributed Systems, Applied Machine Learning, Database Systems

Pune Institute of Computer Technology **Aug 2016 – May 2020**
Bachelor of Engineering in Computer Engineering — ***GPA: 3.96*** *Pune, Maharashtra, India*
Coursework: Data Structures & Algorithms, Operating Systems, Artificial Intelligence, Computer Networks, Discrete Mathematics

Projects

RainStorm – Stream Processing Framework | *Go, Distributed Systems, TCP/UDP Sockets, Design Patterns, System Design*

- Designed RainStorm, a real-time stream processing framework inspired by **Storm** and **Spark Streaming**.
- Developed exactly-once processing semantics using persistent, append-only **Distributed File System** based logging and dynamic task rescheduling, ensuring 100% accuracy during simulated failures of 3 nodes over 5+ runs.
- Benchmarked RainStorm against **Spark Streaming**, achieving equivalent performance for 10000 records.

HyDFS – Distributed File System | *Go, Distributed Systems, Cassandra, TCP/UDP Sockets, Design Patterns, System Design*

- Engineered a DHT-based Distributed File System in **Go**, inspired by **Cassandra** and **HDFS**, employing consistent hashing to achieve sub-5-second replication latencies for files up to 100MB.
- Leveraged quorum replication to tolerate up to 3 simultaneous failures, maintaining re-replication times under 5 seconds.
- Achieved read-my-writes consistency to guarantee immediate visibility of updates during 10000 concurrent client operations.
- Leveraged **SWIM Membership Protocol** to proactively detect node failures in less than 4 seconds.
- Developed a decentralized merge mechanism using unique sequence identifiers for totally ordered merges across all replicas.

Agent Village | *Python, FastAPI, Supabase, LLM, System Design*

- Architected a trust boundary layer at the DB query level across 3 data contexts, eliminating private memory leakage entirely.
- Designed a proactive behavior with a five-factor readiness scoring algorithm, guaranteeing at least one action per 5-minute tick.
- Developed a queue-based inference architecture decoupling LLM calls, resulting in system to scale to 1,000+ agents.
- Built agent observability primitives with token/latency telemetry and trust context logging, reducing mean debug time.

Drop A Doc | *Python, Flask, React, PostgreSQL, Machine Learning, Docker, NLP*

- Drop A Doc is a document-organizing app built in **Android** that automatically categorizes your documents.
- Created **NLP-driven** key-entity extraction model using **Bi-directional LSTM** in **PyTorch** and achieved an accuracy of 92%.
- Reduced the processing time by about 60% & cost by 73% by optimizing processing using On-Device OCR.